PSAT: A PAIRWISE TEST DATA GENERATION TOOL BASED ON SIMULATED ANNEALING ALGORITHM

GOH GHEE HAU

BACHELOR DEGREE OF COMPUTER SCIENCE
(SOFTWARE ENGINEERING)
UNIVERSITI MALAYSIA PAHANG

ABSTRACT

In this information technology era, there is a huge influence of high technology and artificial intelligence when creating new software products in the whole world. To bring high quality software products to the end user, software testing plays important roles and need to be considered during the development stage. Therefore, there is impossible to cover all the test case of the software products and may lead to exhaustive testing. This research is about the research on developing a Pairwise Test Data Generation Tool based on Simulated Annealing (SA) algorithm which named as PSAT. PSAT is used to generate the sufficient test case to reduce the financial resources and time. In PSAT, we using pairwise testing techniques which each interaction of test case considers of two input parameters. Each parameter will have different parameter values that entering by users. In development of PSAT, a prototype of Graphics User Interface (GUI) will be designed and create for user to enter the number of parameters and number of values for each parameter. In this research, the test case will be generated based on the SA algorithm to show that the test case can be generated with sufficient.

ABSTRAK

Dalam era penuh dengan teknologi maklumat ini, terdapat pengaruh besar terhadap teknologi tinggi dan kecerdasan buatan semasa membuat produk perisian baru di seluruh dunia. Untuk membawa produk perisian yang berkualiti tinggi kepada pengguna akhir, ujian perisian memainkan peranan yang penting dan perlu dipertimbangkan semasa peringkat pembanguanan. Oleh sebab itu, adalah mustahil untuk merangkumi semua kes ujian terhadap produk perisian dan ini boleh menyebabkan ujian lengkap terhadap sesuatu produk. Kajian ini adalah mengenai pembangunan terhadap satu *Pairwise Test Data Generation Tool* berdasarkan *Simulated Annealing (SA)* algoritma dengan beri nama *PSAT*. *PSAT* digunakan untuk menjana kes ujian supaya dapat mengurangkan kegunaan sumber kewangan and masa. Dalam *PSAT*, kita menggunakan teknik ujian dari segi pasangan di mana setiap interaksi kes ujian mempunyai dua *parameter input*. Dalam setiap *parameter* akan mempunyai nilai-nilai yang berbeza yang akan dimasukkan oleh pengguna. Dalam pembangunan *PSAT*, satu prototaip *Graphics User Interface (GUI)* akan direka bentuk dan membolehkan pengguna memasukkan nombor *parameter* dan niali bagi setiap *parameter*. Dalam kajian ini, kes ujian dijana mengikut *SA* algoritma untuk menujukan kes ujian dapat dijana dengan mencukupi.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACA	Ant Colony Algorithm
AETG	Automatic Efficient Test Generator
BA_PTC	A Bat-Inspired Strategy for Pairwise Testing With Constraints Support
GA	Genetic Algorithm
GUI	Graphics User Interface
HS	Harmony Search Algorithm
IPO	In-Parameter-Order
IPOG	In-Parameter-Order-General
IPOG-D	In-Parameter-Order-General with D-construction
IPOG-F	In-Parameter-Order-General with FireEye
IRPS	Intersection Residual Pair Set
JDK	Java Development Kit
LAHC	Late Acceptance Hill Climbing Algorithm
OPAT	One Parameter at a Time
OTAT	One Test at a Time
OS	Operating System
PICT	Pairwise Independent Combinatorial Testing
PSAT	Pairwise Test Data Generation Tool Based on Simulated Annealing Algorithm
RAD	Rapid Application Development
RAM	Random Access Memory

R&D	Research and Development
SA	Simulated Annealing Algorithm
SDLC	Software Development Life Cycle
USB	Universal Serial Bus
XP	Extreme Programing

CHAPTER 1

INTRODUCTION

1.1 Introduction

In the information technology era, there is a huge influence of high technology and artificial intelligence when creating new software products in the whole world. This brand new method provides an effective way from bringing high quality software products to the end user. Many fields also rely on this method, especially in the Research and Development (R&D) area. As an evidence, there are many manual processes has been taking place by certain software products or artificial intelligence. Basically, every created product is operating by the combination of hardware and software to implement each feature (Perrouin G. et al. 2011). There is a closely relationship between hardware and software, both are playing an important roles to avoid failure exists.

Failure of products will happen when a human action produces some error or bug in the software and this will lead to the defects which will cause a failure occur when executed. This problem will cause serious damage of system function, and will involve higher cost and loss of time especially for a critical system. Therefore, software testing takes first priority in any Software Development Life Cycle (SDLC) to make sure the quality of software and to prevent the failure of the software.

Software testing defined as the process of executing a program on finding possible errors and validating the software or system against its specification (Myers, G. J., Badgett, T., & Sandler C. 1979). From the studies of seven principles, we know that exhaustive testing is impossible to execute all the test cases for a real software product (Wang, S., Ali, S., & Gotlieb, A. 2013). A complete testing or test for everything is impossible because there are many possible combinations of inputs and pre-condition test case for software.

Pairwise testing is an effective combinatorial method used to minimize the number of the test case that needs to inputs to a system which interactions between two input parameters values (McCaffrey, J. D. 2010). This strategy will be generating test cases that cover all the possible combinations to include the test data and to reduce the possibilities of faults due to interaction (Perrouin, G. et al. 2011). There are many pairwise testing strategies are available in the industry such as Ant Colony Algorithm (ACA) (Shiba, T., Tsuchiya, T. & Kikuno, T. 2004), Automatic Efficient Test Generator (AETG) (Cohen, D. M., Dalal, S. R., Fredman, M. L. & Patton, G. C. 1997), Genetic Algorithm (GA), Harmony Search (HS) Algorithm, In-Parameter-Order (IPO) (Lei, Y. & Tai, K. C. 1998), Intersection Residual Pair Set (IRPS) (Younis, M. I., Zamli, K. Z. & Isa, N. A. M. 2008), Simulated Annealing (SA) algorithm, All-Pairs, and so on. In Chapter 2, several existing strategies such as AETG, GA, IPO and IRPS will be elaborated in details.

1.2 Problem Statement

The main outcome for the software testing is finding defects of the existing software product. Table 1 shows the features on a laptop which consist of different Operating System (OS), Processors, System Type, Random Access Memory (RAM), Graphics Card, Battery, Universal Serial Bus (USB), Hard Drives, Screen Resolutions and Keyboard. Based on Table 1, there are consists of 10 parameters, which each parameter have two values. Therefore, exhaustive testing are happen due to 2 x 2 x 2 x 2 x 2 x 2 x 2 x 2 x 2 x 2 = 2^{10} = 1024 possible combinations need to be test to cover all the test cases. If one test case needs 5 minutes to be tested, the total minutes to complete the test will be 5120 minutes or approximately using 3 days to complete all test cases.

Table 1 Parameters and Values of Laptop Features

Parameters	Values	
OS	Windows	Linux
PROCESSORS	Intel	AMD
SYSTEM TYPE	64 bits	32 bits
RAM	2 GB	4 GB
GRAPHICS CARD	Yes	No
BATTERY	Built In	External
USB	2.0	3.0
HARD DRIVES	500 GB	1 TB
SCREEN RESOLUTION	1024×768	1280×800
KEYBOARD	With Number Pad	Without Number Pad

1.3 Goal and Objectives

The main goal of this research is to develop a Pairwise Test Data Generation Tool based on Simulated Annealing algorithm which given a name as PSAT. Following are the several objectives to achieve the goal:

i) To study the pairwise testing generation by reviewing the existing pairwise testing strategies.
ii) To apply Simulated Annealing (SA) algorithm into PSAT.
iii) To evaluate and compare the performance of proposed PSAT against with other existing strategies in term of test size.

1.4 Scope

The PSAT will be developing by using the NetBeans8.0.2 with JFrame and Java Development Kit (JDK) 8.0. Following are the scope of the research:

i) A pairwise tool with design of Graphics User Interface (GUI).
ii) A pairwise tool that adopting SA algorithm.
iii) A pairwise tool that consisting specifies values for each parameter.

1.5 Thesis Organization

This research consists of five (5) chapters that discuss the main detail in each chapter. Chapter 1 is Introduction. This chapter discusses the research introduction that will be done with including the problem statement, goal, objectives, and scope for PSAT.

Chapter 2 is Literature Review. In this chapter, we will discuss the existing research and literature review that related to the research.

Chapter 3 is Methodology. The overall approaches and framework of the research will be discussed detail in this chapter. The method, techniques or approaches will be shown on this chapter also.

Chapter 4 is Design, Implementation and Result Discussion of PSAT. In this chapter, a PSAT design will be illustrated based on the selection algorithm and implemented into PSAT. The detail of the implementation process that involved will be discussed. Lastly, the final test cases will be analyzed and evaluated whether the development is success or failure. The result also used to compare with other strategies.

Chapter 5 is Conclusion. All research that has done will be summarizing in this chapter. Besides that, the future work and alternative way to improve the research also state in this chapter.

1.6 Summary

This chapter discussed the introduction of the research of PSAT: Pairwise Test Data Generation Tool by adopting the Simulated Annealing (SA) algorithm. Problem statement, goals and objectives, scope and thesis organization are the content of this chapter. Next chapter will discuss the existing pairwise testing strategies that should be review to development our PSAT.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In the Chapter 1, we discuss about the research introduction which included the problem statement, goals and objectives, and scope. In this Chapter, survey of the relevant literature review will be carried out to gain idea to design the PSAT. In particular, the pairwise testing techniques and existing pairwise testing strategy will be elaborated to justify the current work.

2.2 Overview

There are many existing strategy has been published and released to industry to be used in different field. In order to achieve the goal of this research, pairwise testing will be discussed in details.

2.2.1 Pairwise Testing

Pairwise testing is an effective combinatorial method used to minimize the number of the test case that needs to inputs to a system which interactions between two input parameters values (McCaffrey, J. D. 2009). In pairwise testing, a test suite will be generated to covers all the possible combination that has consists of the test data values for each pair of parameters. To understand the pairwise testing, an example will be shown with a system with four parameters which is Router, Browser, Web Server and Database Server. Each parameter of the system will consist of different values such as Router has Cisco and Huawei, Browser has Internet Explorer and Google Chrome, Database Server has Oracle and SQL Server and Web Server has Apache and Jboss as values as shown in Figure 1.

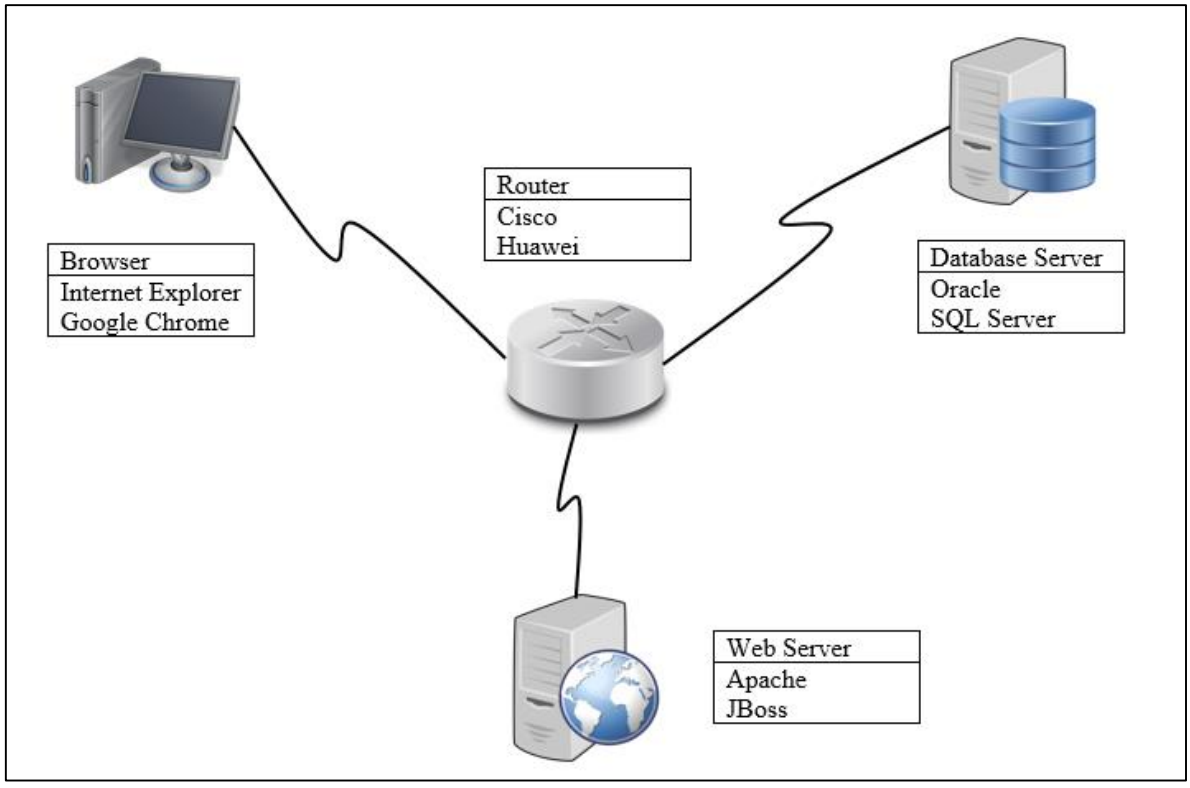

Figure 1 A System Consists of Four Parameters

Based on Figure 1, we assign the input variables into unknown term which are consists of combination of alphabet and numeric number. The result is shown on Table 2.

Table 2 Input Variables and Values of Four Parameters

Base Values	**Input Variables**			
Parameter	Router (P1)	Browser (P2)	Database Server (P3)	Web Server (P4)
Parameter Values	Cisco (A1)	Internet Explorer (B1)	Oracle (C1)	Apache (D1)
	Huawei (A2)	Google Chrome (B2)	SQL Server (C2)	Jboss (D2)

From the Table 2, the four parameters which are Router, Browser, Database Server and Web Server are assigning as Parameter 1 (P1), Parameter 2 (P2), Parameter 3 (P3) and Parameter 4 (P4). The values of each parameter also are assigning with the term of A1, A2, B1, B2, C1, C2, D1 and D2 which is combination of alphabet and numeric number. The four inputs variables are consist of two selections from the system respectively. In this situation, there are $2^4 = 16$ exhaustive combinations to cover all the possible test data and the result is shown in Table 3.

Table 3 Exhaustive Combinations of P1, P2, P3 and P4

Base Values	Input Variables			
Parameters	P1	P2	P3	P4
Parameter Values	A1	B1	C1	D1
	A2	B2	C2	D2
Exhaustive Combinations				
T1	A1	B1	C1	D1
T2	A1	B1	C1	D2
T3	A1	B1	C2	D1
T4	A1	B1	C2	D2
T5	A1	B2	C1	D1
T6	A1	B2	C1	D2
T7	A1	B2	C2	D1
T8	A1	B2	C2	D2
T9	A2	B1	C1	D1
T10	A2	B1	C1	D2
T11	A2	B1	C2	D1
T12	A2	B1	C2	D2
T13	A2	B2	C1	D1
T14	A2	B2	C1	D2
T15	A2	B2	C2	D1
T16	A2	B2	C2	D2

Pairwise testing techniques have been used to reduce the exhaustive combinations for this system. By using this technique, a 2-way possible combinations produce P1P2, P1P3, P1P4, P2P3, P2P4 and P3P4 as the combinations. For P1P2 combination, the test case has been reduced to four test cases which only consider P1 and P2 as a pair. P3 and P4 are randomly assigned a value as shown in Table 4.

Table 4 2-way Combination for P1P2

Base Values	**Input Variables**			
	P1	P2	P3	P4
	A1	B1	C1	D1
	A2	B2	C2	D2
2-way Combinations for P1P2				
Test Case 1	A1	B1	C1	D1
Test Case 2	A1	B2	C2	D2
Test Case 3	A2	B1	C1	D1
Test Case 4	A2	B2	C2	D2

The second combination is P1P3. For this combination, the test case also been reduced become four test cases and only consider P1 and P3 as a pair. P2 and P4 are randomly assigned a value as shown in Table 5.

Table 5 2-way Combination for P1P3

Base Values	**Input Variables**			
	P1	P2	P3	P4
	A1	B1	C1	D1
	A2	B2	C2	D2
2-way Combinations for P1P3				
Test Case 1	A1	B1	C1	D1
Test Case 2	A1	B2	C2	D2
Test Case 3	A2	B1	C1	D1
Test Case 4	A2	B2	C2	D2

The third combination is P1P4. For this combination, the total test size also reduced become four test cases and only considers P1 and P4 as a pair. P2 and P3 are randomly assigned a value as shown in Table 6.

Table 6 2-way Combination for P1P4

Base Values	Input Variables			
	P1	P2	P3	P4
	A1	B1	C1	D1
	A2	B2	C2	D2
2-way Combination for P1P4				
Test Case 1	A1	B1	C1	D1
Test Case 2	A1	B2	C2	D2
Test Case 3	A2	B1	C1	D1
Test Case 4	A2	B2	C2	D2

The fourth combination is P2P3. For this combination, the total test size also reduced become four test cases and only considers P2 and P3. The value is randomly assigned for P1 and P4 as shown in Table 7.

Table 7 2-way Combination for P2P3

Base Values	Input Variables			
	P1	P2	P3	P4
	A1	B1	C1	D1
	A2	B2	C2	D2
2-way Combination for P2P3				
Test Case 1	A1	B1	C1	D1
Test Case 2	A1	B1	C2	D2
Test Case 3	A2	B2	C1	D1
Test Case 4	A2	B2	C2	D2

The fifth combination is P2P4. For this combination, the total test size also reduced become four test cases and only considers P2 and P4. The value is randomly assigned for P1 and P3 as shown in Table 8.

Table 8 2-way Combination for P2P4

Base Values	Input Variables			
	P1	P2	P3	P4
	A1	B1	C1	D1
	A2	B2	C2	D2
2-way Combination for P2P4				
Test Case 1	A1	B1	C1	D1
Test Case 2	A1	B1	C2	D2
Test Case 3	A2	B2	C1	D1
Test Case 4	A2	B2	C2	D2

Lastly, the combination is P3P4. For this combination, the total test size also reduced become four test cases and only considers P3 and P4. The value is randomly assigned for P1 and P2 as shown in Table 9.

Table 9 2-way Combination for P3P4

Base Values	Input Variables			
	P1	P2	P3	P4
	A1	B1	C1	D1
	A2	B2	C2	D2
2-way Combination for P3P4				
Test Case 1	A1	B1	C1	D1
Test Case 2	A1	B2	C1	D2
Test Case 3	A2	B1	C2	D1
Test Case 4	A2	B2	C2	D2

After merging the combinations of P1P2, P1P3, P1P4, P2P3, P2P4 and P3P4, the similar interaction of the combinations are created as a new test case and save into the final test suite. After save into the final test suite, the interaction are deleted and do not count for the next test case until all the interaction is covered. The result of this system shows that the final test suite has been reducing from 16 test cases to 8 cases which reduction of 50%. Figure 2 shown the total test cases after merge all the combinations.

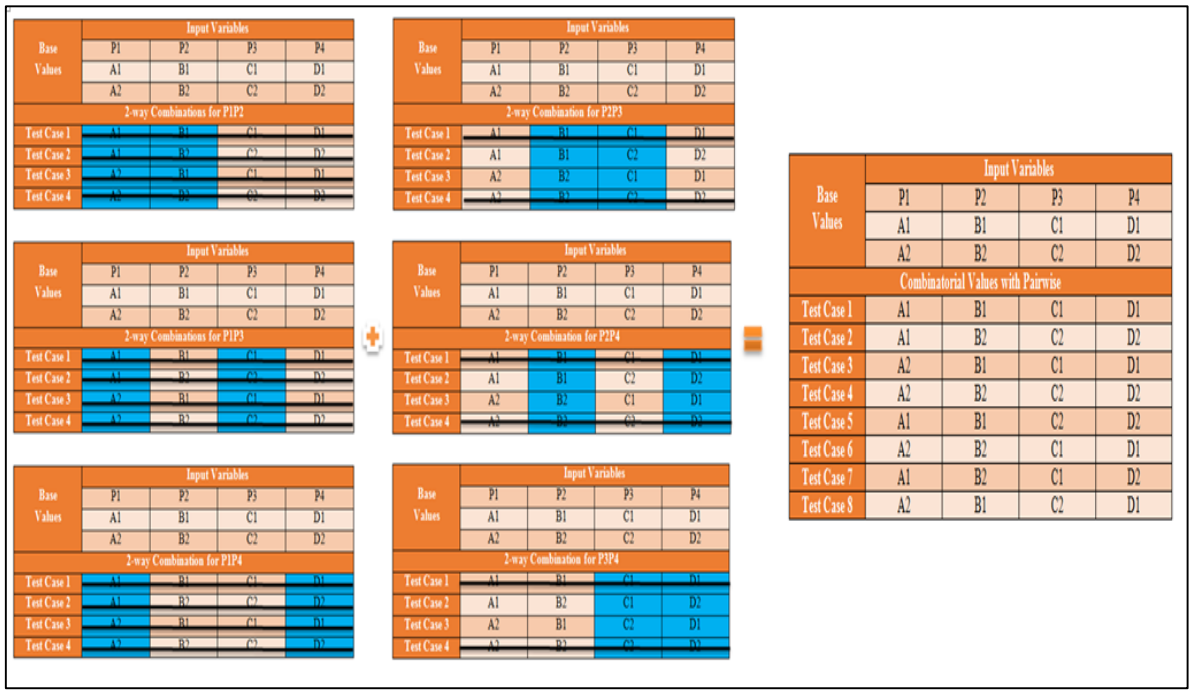

Figure 2 Merging of the P1, P2, P3 and P4

2.3 Survey of Existing Pairwise Strategies

In this topic, the surveys will be done based on the existing work on pairwise testing. This existing pairwise testing is difference in two approaches which are one test at a time (OTAT) and one parameter at a time (OPAT). We will review some existing techniques for each approach. In one test at a time approaches, we will review the GA (Mitchell, M. 1995) and AETG (Cohen, D. M., Dalal, S. R., Fredman, M. L. & Patton, G. C. 1997). While in one parameter at a time approaches, we will review on IPO (Lei, Y. & Tai, K. C. 1998) and IRPS (Younis, M. I., Zamli, K. Z. & Isa, N. A. M. 2008). Figure 3 show the design of the existing pairwise strategies.

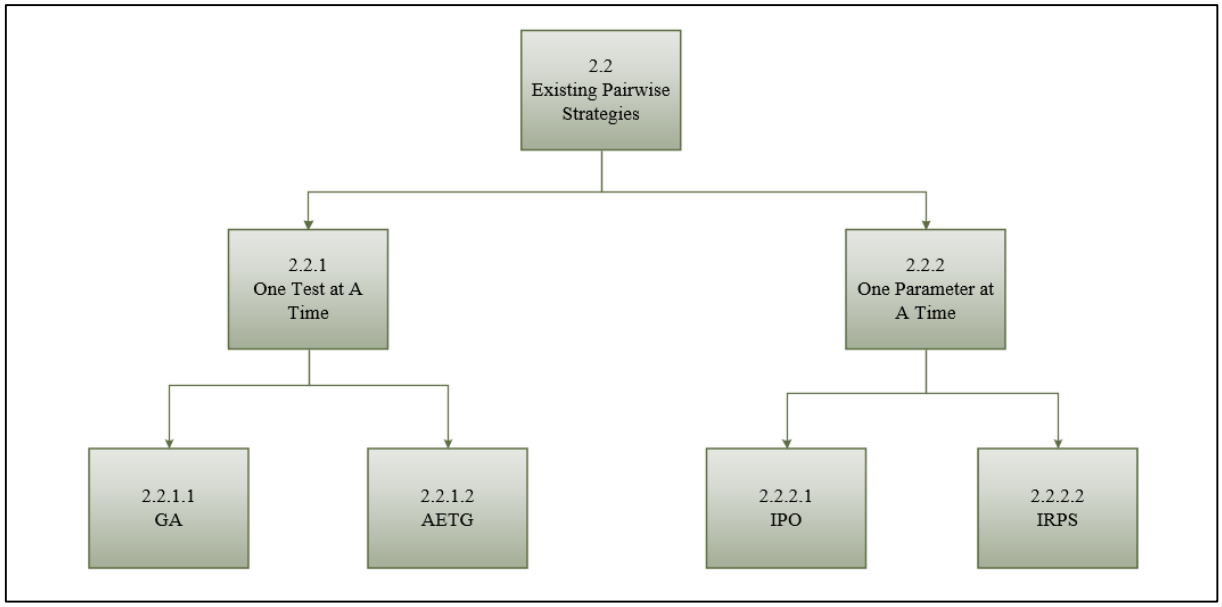

Figure 3 Existing Pairwise Strategies